ORNL/TM-2014/670

Evaluating Storage Systems for Lustre

Sarp Oral

August 20, 2015

Approved for public release; distribution is unlimited

OAK RIDGE NATIONAL LABORATORY

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-2014/670

National Center for Computational Sciences

Evaluating Storage Systems for Lustre

Sarp Oral

Date Published: August 20, 2015

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831-6283
managed by
UT-BATTELLE, LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

ABSTRACT

Storage systems are complex, including multiple subsystems and components. Sustained operations with top performance require all these subsystems and components working as expected. Having a detailed performance profile helps establishing a baseline. This baseline can be used for easier identification of possible future problems. A systematic bottom-to-top approach, starting with a detailed performance analysis of disks and moving up across layers and subsystems, provides a quantitative breakdown of each component's capabilities and bottlenecks. Coupling these low-level tests with Lustre-level evaluations will present a better understanding of performance expectations under different I/O workloads.

1. END-TO-END LUSTRE I/O PERFORMANCE PROFILING

I/O performance profiling is the systematic approach for identifying a system's top-level aggregate performance and performance capabilities and shortcomings of its components or subsystems. The end-to-end system will categorically include file system clients, file system servers, back end storage disk subsystem, client to server system area network and server to back end disk storage area network. Identifying performance profiles of each of these will help ensuring the overall system's initial deployment is performed according to the specifications. It will also help maintaining the operational performance for the lifetime of file system.

Oak Ridge Leadership Computing Facility (OLCF) at Oak Ridge National Laboratory (ORNL) has assembled a series of scripts and synthetic benchmarks that comprise a benchmark suite to better assess the server side of file and storage systems for Lustre file system under OLCF specific I/O workloads [**1**]. This benchmark suite is open source and made available to public can be obtained directly from OLCF.

The purpose of the benchmark suite is to establish the performance profile of the various file and storage system technologies by executing a series of scripts and synthetic benchmarks representative of common I/O workloads observed at OLCF production file systems.

2. OLCF LUSTRE BENCHMARKING

I/O performance profiling at OLCF includes multiple separate efforts, using various tools for I/O benchmarking. These efforts target analyzing I/O performance at various layers and subsystems and they complement one another. Of these, OLCF Scalable Storage System (SSS) benchmark suite profiles the server side I/O performance, OLCF LNET tests evaluate the Lustre networking layer performance between clients and servers, and a multitude of synthetic benchmarks and real-world scientific applications assess the file system performance at the Lustre client level. Client and Lustre networking level performance evaluations are out of scope of this document.

2.1 OLCF SCALABLE STORAGE SYSTEM (SSS) BENCHMARK SUITE

The OLCF Scalable Storage System (SSS) benchmark suite focuses of profiling the server-side I/O performance and is composed of 2 parts with the following goals:

1. A block-level I/O benchmark suite can be used to establish and evaluate the performance of the block-level storage and provide a performance scalability profile of the storage system with respect to I/O block sizes between 4 kilobytes and 8 megabytes, to the number of exported LUNs per host, and the number of hosts for both random and sequential I/O patterns.

2. A Lustre file system-level I/O benchmark suite can be used to establish and evaluate the performance scalability profile of the file system with respect to transfer (record) size between 4 kilobytes and 7 megabytes, and the number of Lustre object storage targets (OSTs) (i.e. LUNs) per host, and the number of object storage servers (OSSs) (i.e. hosts).

The duration of the benchmark suites will depend on the configuration of the test system.

1. The block-level benchmark is time-constrained will take approximately 22 hours, plus 2.8 hours for each LUN assigned to a single host under test. For example, if each host in the test system has 7 LUNs assigned to it, the block-level benchmark suite will take approximately 42 hours for each offered configuration.

2. The file system-level benchmark is data constrained. As a reference point, executing the file system-level benchmark on a test system capable of 10 GB/s aggregate bandwidth, with 4 hosts each driving 5 LUNS (8+2 RAID6, SATA drives) requires approximately 24 hours to complete.

OLCF is interested in top-level and sustainable performance of the system under typical operating conditions. The benchmark suite is designed to drive the sustainable performance. Both healthy and degraded mode performances are equally important.

The block-level benchmarks can first be executed with the storage system running in a healthy state. “Healthy mode” means that all LUNs that are exercised during the block-level tests are in optimal conditions.

Additional tests can be performed to demonstrate performance in a degraded mode. “Degraded mode,” means, according OLCF’s operational guidelines, that at least 10% of the LUNS that are exercised during the block-level tests are being rebuilt for the entire duration of the test.

If the storage systems tracks areas of a LUN that have been written such that it could respond to a read request to an unwritten area of the LUN without accessing the disk drives, this ability can be disabled or defeated for the duration of the benchmark execution.

In order to run the OLCF benchmark suite, before benchmarking, nodes in the test system will need to be configured with passwordless root *ssh* capability. The *pdsh* package is required for benchmarks to synchronize, run, and gather data.

For more realistic evaluation for performance evaluation of initial deployments, it is preferred that the benchmarked configurations to be identical to the production configuration, including the operating system, kernel, and Lustre code versions. For example, if parity declustering or write cached mirroring will be used in production configuration, these features must be enabled during the execution of the benchmarks.

For ensuring proper data provenance, all changes to benchmark suite must be documented and recorded with the data itself. Benchmark suite collects system configuration information as much as possible and catalogues this along with the test data.

2.2 BLOCK-LEVEL BENCHMARKS

The block-level benchmark suite measures system performance under different configurations and I/O patterns. The benchmark suite consists of multiple benchmarks that will iteratively profile the storage system performance. It is preferred that DM-Multipath be configured for communication with the Test system. More details on DM-Multipath and its configuration may be found at [**2**].

The test engine used in the benchmark suite to exercise block device targets is a synthetic benchmark named *fair-lio*, developed by ORNL. The source code this benchmark is publicly available. For instructions on building and running this benchmark, please refer to *README* file provided in the source code distribution. *Fair-lio* uses the *libaio* library provided by the *libaio-devel* package; this package is required to build the benchmark engine. Configure the test system and install the required packages for the Test system OS distribution and kernel prior to building the *fair-lio* benchmark.

There are three test scripts in the healthy mode block-level benchmark suite. These are: *block-io-single-host-full-run.sh*, *block-io-single-host-scale-up.sh*, and *block-io-ssu-scale-up.sh*. Run each these three scripts separately on the test system in healthy mode. Each of these scripts will execute a series of randomized (with respect to number of devices to be tested, block sizes, queue depth, and individual test iterations) set of tests. Of these three scripts, the *block-io-single-host-full-run.sh* will exercise and profile the performance of a single LUN on a single test host, while the *block-io-single-host-scale-up.sh* script will exercise and profile the performance of all LUNs on a single host. The *block-io-ssu-scale-up.sh* script will use all LUNs on all test hosts and provide the performance profile of the SSU.

There is one test script in the degraded mode block-level benchmark suite. The *block-io-ssu-degraded.sh* script will exercise all LUNs on all test hosts and provide the performance profile of the SSU when 10% of the LUNs are being rebuilt. Document all LUNs on the test hosts, clearly indicating degraded LUNs and rebuild start times. Further, ensure that a minimum of 10% of the LUNs is in active rebuild state for the entire execution of the *block-io-ssu-degraded.sh* script.

A “completed block I/O suite” means that all three scripts are executed on a healthy system and all scripts ran to completion. Further, the *block-io-ssu-degraded.sh* script is also executed to completion on a degraded system.

For more details on configuring and running the block I/O benchmark suite, please refer to the *README* file included the benchmark suite package.

2.3 BLOCK-LEVEL BENCHMARK OUTPUT

Each test script included in the benchmark suite will write a temporary randomized test list file, a results summary output file, a comma separated detailed results and statistics file. Each test script will also generate a raw test output directory and write individual test results to a separate file in this directory.

Tar the contents of the *blockio-test-output* directory and all results of all tests of all iterations, as well as all *stdout* and *stderr* messages with accurate time stamps for post-processing and long-term archival of the data generated.

2.4 LUSTRE-LEVEL BENCHMARKS

The Lustre-level benchmark suite consists of three files: *run_obdfilter_survey.sh*, *obdfilter-survey*, and a *README* file. The *obfilter-survey* script provided in the ORNL OLCF-3 SSS file system-level benchmarks suite differs from the one provided by the Lustre IOKIT. The *run_obdfilter_survey.sh* is the main benchmark script, and will drive the ORNL provided *obdfilter-survey* script with the proper parameters.

OLCF provided a custom designed *obdfilter-survey* script specific to OLCF workload.

A Lustre file system is required prior to running the *run_obdfilter_survey.sh* script. Configure and build a Lustre file system (identical to the version to be used in production) on the test system prior to running the file system-level benchmarks. There is detailed information provided in configuring and building a Lustre file system in Lustre wiki documents (*http://wiki.lustre.org/*).

Configure and build the Lustre file system to obtain maximum aggregate performance per OSS and maximum aggregate performance for all OSSs.

A Lustre file system in the context of OLCF SSS file system-level benchmarks is defined as the collection of Lustre Object Storage Servers (OSSs), a Lustre Metadata Server (MDS), a Lustre Management Server (MGS). There are no requirements to have separate nodes configured as Lustre clients to run the *run_obdfilter_survey.sh* script.

Lustre OSS nodes need to be configured before running the *run_obfilter- survey.sh* script. Check the output of "*lctl dl*" command on the OSS nodes to verify the existence of *obdfilter* instances.

Modify the *archive_name* and *raw_values* variables in the *run_obdfilter_survey.sh*. Detailed instructions on how to modify these two variables are included in the *README* file distributed in the OLCF file system-level benchmark suite.

Modify the *run_obdfilter_survey.sh* script to provide the required and correct listing of OST devices for operation prior to running the script.

As a good practice for governing data provenance, tar the contents of the *$archive_home/completed_sets* directory and record all results of all tests of all iterations, as well as all *stdout* and *stderr* messages with accurate time stamps.

ACKNOWLEDGEMENTS

This research used resources of the Oak Ridge Leadership Computing Facility, located in the National Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the Department of Energy under Contract DE-AC05-00OR22725.

This manuscript has been authored by UT-Battelle, LLC, under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

BIBLIOGRAPHY

[1] Sarp Oral. (2012, December) OLCF Spider 2 RFP. [Online]. http://www.olcf.ornl.gov/ wp-content/uploads/2010/03/olcf3-benchmark-suite.tar.gz

[2] RedHat. DM Multipath. [Online]. https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_ Linux/6/html/DM_Multipath/